

# lab4 report

---

于新雨 计25 2022010841

## 功能实现

### sys\_linkat

首先检查新旧文件名是否相同，如果是则返回错误

在当前任务的内部数据结构中维护了一个 links 向量，存储了(新链接名, 原文件名)的元组，我们查找和现有文件名对应的文件描述符，如果有，则更新 links 向量，插入新的元组，并且增加 `nlinks` 引用计数

### sys\_unlinkat

首先查找该链接名对应的真实路径，如果找到，则进一步查找是否有对应的打开的文件描述符

然后从 links 列表中删除指定的链接项，并且更新文件的 nlink

如果某个真实路径的所有链接都被删除 (`link_cnt == 0`)，则调用 `remove_inode` 删除实际文件

### fstat

我们检查 `_fd` 是否超出当前任务的 `fd_table` 范围，并且克隆文件对象

然后调用 `file.fstat()` 获取文件的元数据，将结果写入用户空间的 `Stat` 结构（通过 `virt_to_phys` 转换虚拟地址）

## 问题回答

### ch6 问题回答

在我们的easy-fs中，root inode起着什么作用？如果root inode中的内容损坏了，会发生什么？

inode 结构体是表示文件或者目录的基本数据结构，root inode 表示的是文件系统的根目录，所有的文件和目录都在这个 inode 下

inode 的结构如下所示

```
pub struct Inode {  
    /// getting the block id  
    pub block_id: usize,  
    /// getting the offset of the block  
    pub block_offset: usize,  
    fs: Arc<Mutex<EasyFileSystem>>,  
    block_device: Arc<dyn BlockDevice>,  
}
```

如果 root inode 中的内容损坏了，文件系统将无法访问根目录下的文件和目录，导致整个或者部分文件系统不可用

### ch7 问题回答

举出使用 pipe 的一个实际应用的例子

像是 cmdline 里面的 pipe symbol |，可以将一个命令的输出作为另一个命令的输入

一个常见的用法是输出一大段文本到 grep 进行过滤

如命令 `ROPgadget --binary ./challenge --only "add|ret"|grep rdi|grep rax`，该命令用 `ROPgadget` 查找二进制文件中的 ROP gadget，输出较多，然后通过两次 pipe 给 grep 使得输出中只保留包含 rdi 和 rax 的行

如果需要在多个进程间互相通信，则需要为每一对进程建立一个管道，非常繁琐，请设计一个更易用的多进程通信机制

可以用共享内存（shm）来实现多个进程间通信，把同一个物理内存区域映射到不同的进程地址空间中，所有进程都可以访问这个共享内存区域，从而可以通过读写该区域来通信

也可以使用消息队列来实现多个进程间通信，消息队列是一个链表结构，允许多个进程以 FIFO 的方式进行通信，进程可以向消息队列中发送和接收消息，消息队列的优先级可以设置，从而实现不同优先级的进程之间的通信

## honor code

在完成本次实验的过程（含此前学习的过程）中，我曾分别与 以下各位 就（与本次实验相关的）以下方面做过交流，还在代码中对应的位置以注释形式记录了具体的交流对象及内容：

无

此外，我也参考了 以下资料 ，还在代码中对应的位置以注释形式记录了具体的参考来源及内容：

无

我独立完成了本次实验除以上方面之外的所有工作，包括代码与文档。我清楚地知道，从以上方面获得的信息在一定程度上降低了实验难度，可能会影响起评分

我从未使用过他人的代码，不管是原封不动地复制，还是经过了某些等价转换。我未曾也不会向他人（含此后各届同学）复制或公开我的实验代码，我有义务妥善保管好它们。我提交至本实验的评测系统的代码，均无意于破坏或妨碍任何计算机系统的正常运转。我清楚地知道，以上情况均为本课程纪律所禁止，若违反，对应的实验成绩将按“-100”分计